Assignment -3 Report
submitted for Artificial
Intelligence (UNC504)
by

Name of student: Prabhnoor Singh
Roll number: 102115059 Group: 3NC3

Submitted to

# Dr. Tanvi Dovedi

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, (A DEEMED TO BEUNIVERSITY), PATIALA, PUNJAB
INDIA

July-Dec 2023

QI. Solve 8-D puzzle using A* algorithm and print the shortest path.

Sol.

```
import heapq

graph = {
'S1': [('A', 2), ('B', 4), ('C', 4)],
'A': [('D', 3), ('E', 7), ('G', 9)],
'B': [('G', 4)],
'C': [('G', 5)],
'D': [],
```

```
    o, '
E ' : o,
'G': o,

def heuristic(node, goal):
    return      # Simple heuristic function for this example

def astar(graph, start, goal): open_list = [ (O' start)]
    came_from = { } g_score = {node: float( ' inf' ) for
    node in graph} g_score[start] - 0f_score = {node:
    float( 'inf') for node in graph} f_score[start] =
    heuristic(start, goal)

    while open_list:
        current = heapq. heappop (open_list) if
        current goal :
            return reconstruct_path(came_from, current) for
        neighbor, cost in graph[current] :
                                                + cost if tentative_g_score < :
    came_from[neighbor] = current g_score[neighbor] = tentative_g_score
    f_score[neighbor] = + heuristic (neighbor, goal) heapq. heappush (open_list,
    neighbor) ) return None

def reconstruct_path(came_from, current) :
    path — [current] while current in came
    from: current = came_from[current] path.
    insert (O, current) return path

start node = 'SI goal_node — 'G'optimal_path —
astar(graph, start_node, goal_node) if optimal_path:
    print( "Optimal Path: % optimal_path)
else :
    print("No path found. ")
```

Optimal Path: [ • SI','B',